

REMARKS/ARGUMENTS

Claims 1- 13 are pending in the present application. The specification has been amended to correspond to the drawings as suggested by the Examiner. Claim 2 has been amended to provide antecedent basis as suggested by the Examiner. Claims 10 and 13 have been amended to more clearly describe the invention. No new matter has been added.

A marked-up version for the claims being changed is also attached to this paper entitled "**Marked-Up Version Showing Changes Made**".

Specification

The specification stands objected to because "pg. 11 line 20 states memory location 38, this should state memory location 64." (Office Action, page 2). The specification has been amended to correct this error. Thus, it is respectfully asserted that this objection be withdrawn.

Claims

Claim 2 stands objected to because it allegedly "recited the limitation 'second type' in pg. 18 line 2. There is insufficient antecedent basis for this limitation in the claim." (Office Action, page 2). Claim 2 has been amended to correct this error by amending claim 2 to read "wherein said first value indicates that the file will not be deleted upon closing and the second value indicates that the file will be deleted upon closing." Thus, it is respectfully requested that this objection be withdrawn.

35 U.S.C. §102 Rejections

Claims 1, 5, and 9

Claims 1, 5, and 9 stand rejected under 35 U.S.C. §102(b) as allegedly being anticipated by Orita (U.S. Patent No. 5,163,147). This rejection is respectfully traversed.

According to the M.P.E.P. § 2131, "a claim is anticipated [under 35 U.S.C. § 102(b) and (e)] only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *See also Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

Claims 1, 5 and 9 recite the following limitations:

"a first memory associated with the file, said first memory for storing a fixed file security status, said fixed file security status being of a first type;

a second memory associated with the file, said second memory for storing an active file security status."

The Office Action alleges that Orita teaches:

"a first memory (see col. 2 line 68: read/write memory, and Fig. no. 14) associated with the file, said first memory for storing a fixed file security status, said fixed file security status being of a first type (see col. 3 lines 1-5); a second memory associated with the file, said second memory (see col. 2 line 68 and Fig 1 no. 14) for storing an active file security status . .
."

The Office Action equates the first memory and the second memory as the same memory. Applicant respectfully disagrees. Upon a closer review of Orita, Orita does not

teach having a first memory and a second memory associated with the file. Orita merely teaches:

"a work station **10** used as a terminal device, a host computer **11** and an external storage unit **12**. . . . The external storage unit **12** is a hard disk, for example, and is operated under the control of the host computer **11**. . . . The host computer **11** includes a central processing unit (CPU) **13** and a read/write memory (RAM) **14** . . ." (Col. 2, lines 53-68).

Thus, Orita does not teach having a first and second memory associated with a file. Orita merely discloses the use of an external storage unit, such as a hard disk, that is not associated with a file in the host computer. Moreover, Orita does not teach having the memory of the host computer associated with a file.

Additionally, claim 1 further recites "a request handler receiving a request from the client to perform operations on the file, said request handler disallowing the client from performing operations on the file if said active file security status is of said first type and allowing the client to perform operations on the file if said active file security status is of said second type." As further stated in the specification, "routers also function as servers, receiving requests from clients or processes and responding to those requests. Therefore, routers are programmed with routines for appropriately responding to a variety of requests. These routines are referred to herein as request handlers." (Specification, page 3, lines 5-8). Thus, in the claimed invention, the request handler is a routine that is programmed in a router or server, which Orita does not disclose or suggest. In fact, Orita does not teach the use of a router, server, or request handler.

Thus, since each and every element as set forth in the claimed invention is not found, either expressly or inherently described in Orita, it can not be said to anticipate the

claimed invention. Accordingly, it is respectfully requested that this rejection be withdrawn.

Claims 10-13

Claims 10-13 are rejected under 35 U.S.C. 102(e) as being allegedly anticipated by Scott, et al. (U.S. Patent No. 5987123). This rejection is respectfully traversed.

Amended claims 10 and 13 recite:

“receiving from a user an open for write call for a file that does not exist at the time the call is received;
recognizing that the file does not exist at the time the call is received;
creating a file entry for said file.”

As stated in the Specification, this is to allow a user to “creat[e] a new secure file.” (Specification, page 15, line 21).

The Office Action alleges:

"As per claims 10 and 13, Scott teaches a method (see Abstract and col. 1 lines 35-38) and a program storage device readable by a machine, tangibly embodying a program of instructions executable by a machine (see col. 3 lines 14-19), for creating a secure file on a file system (see col. 2 lines 12-13), the method comprising: receiving from a user an open for write call (see col. 4 lines 2-5 and col. 5 lines 46-47) for a file that does not exist at the time the call is received; recognizing that the file does not exist at the time the call is received; creating a file entry for said file; receiving from said user an authorization credential (see col. 5 lines 51-56); authenticating the privileges of the user (see col. 4 lines 5-8 and 39-54); recognizing the combination of a user sending an open for write call for a file that does not exist at the time the call is received and an authorization credential that is authenticated (see col. 4 lines 5-8 and 39-54); and creating a secure file (see col. 5 lines 57-59) (see Fig. 5)."

Upon a closer review of Scott, Scott merely discloses a “file system [that] incorporates two levels of validation for programs and data” to allow “a computer system to trust both program and data files without the intervention of the user and with decreased possibility of circumventing the model of trust.” (Col. 1, lines 35-41). Scott does not teach or disclose “receiving from a user an open for write call for a file that does not exist at the time the call is received; recognizing that the file does not exist at the time the call is received” or “creating a file entry for said file” as claimed in claims 10 and 13.

Although Scott uses the term “creating a secure file” (Col. 2, lines 12-13), as cited in the Office Action, Scott does not teach or disclose the creation of a **new** secure file. As described above, Scott merely provides for the use of a two level validation to create the “secured” files and does not create **new** secure files by “recognizing that the file does not exist at the time the call is received” as claimed in claims 10 and 13.

Thus, since each and every element as set forth in the claimed invention is not found, either expressly or inherently described in Scott, it can not be said to anticipate the claimed invention. Accordingly, it is respectfully requested that this rejection be withdrawn.

Remaining Dependent Claims

The remaining dependent claims depend from independent claims 1, 5, or 10 and include the limitations of their corresponding base claims. The base claims being allowable, the dependent claims must also be allowable.

Summary

Given the above reasons, the cited prior art can not be said to render the claimed invention obvious. In view of the above, it is respectfully asserted that the claims are in condition for allowance.

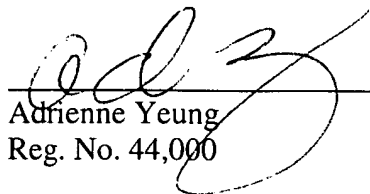
Request for Allowance

It is believed that this Response places the above-identified patent application into condition for allowance. Early favorable consideration of this application is earnestly solicited.

If, in the opinion of the Examiner, an interview would expedite the prosecution of this application, the Examiner is invited to call the undersigned attorney at the number indicated below.

Dated: 7/22/, 2002

Respectfully submitted,
THELEN REID & PRIEST, LLP


Adrienne Yeung
Reg. No. 44,000

THELEN REID & PRIEST LLP
P. O. Box 640640
San Jose, CA 95164-0640
Tel: (408) 292-5800

Marked-Up Version Showing Changes Made

In the Specification:

Paragraph beginning at line 15 of page 11 has been amended as follows:

To change delete-on-close from value "do not delete on close" to value "delete on close", a client 24 requests to open secure file 32. At this point, the server may grant the client exclusive access to the file if the client requests it. A file entry 60 is created, type "operations not allowed" is copied from the fixed security status in memory location 38 to the active file security status in memory location 62, and delete-on-close is initialized to value "do not delete on close" in memory location [38] 64. [A] An authorized credential is passed from the client 24 to the independent verification routine 14. Upon successful validation, the active file security status 62 is changed to access type "operations allowed". A set delete-on-close request is made by the client 24. The server 12 then checks the active file security status 62 in the file entry 60. The server then changes the delete-on-close status in memory location 64 to "delete on close". Then, upon closing, the file 32 is deleted.

In the Claims:

Claims 2, 10 and 13 have been amended as follows:

2. (Once Amended) The apparatus of claim 1, further comprising a third memory associated with the file, said third memory for storing a delete-on-close status, said delete-on-close status initially set to a first value and changeable to a second value,

wherein said first value indicates that the file will not be deleted upon closing and the second value [type] indicates that the file will be deleted upon closing.

10. (Once Amended) A method for creating a secure file on a file system, the method comprising:

receiving from a user an open for write call for a file that does not exist at the time the call is received;

recognizing that the file does not exist at the time the call is received;

creating a file entry for said file;

receiving from said user an authorization credential;

authenticating the privileges of the user;

recognizing the combination of a user sending an open for write call for a file that does not exist at the time the call is received and an authorization credential that is authenticated; and

creating a secure file having a fixed file security status being of a first type.

13. (Once Amended) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform a method for creating a secure file on a file system, the method comprising:

receiving from a user an open for write call for a file that does not exist at the time the call is received;

recognizing that the file does not exist at the time the call is received;

creating a file entry for said file;

receiving from said user an authorization credential;

authenticating the privileges of the user;

recognizing the combination of a user sending an open for write call for a file that does not exist at the time the call is received and an authorization credential that is authenticated; and

creating a secure file having a fixed file security status being of a first type.